

Solved by: Saher (Well Wisher / Aqualeo)

[www.freeittips.com](http://www.freeittips.com)Year 2013**Separate the system window class...(Marks 5)****1: Button****2: DDEMLEvent****3: Message****4: Scroll Bar****5: ComboBox****Answer:**

DDEMLEvent and Message are System window classes.

**Q#2: Separate the Queue and Non-Queue messages?? (Marks 5)****1: Broadcast System Messages****2: GetMessage Extra Information****3: Send Message****4: Get Message Time****5: Send Notify Message****Answer:**

1: Broadcast System Messages : Non-Queued message

2: GetMessage Extra Information : Queued messages

3: Send Message : Non-Queued message

4: Get Message Time : Queued message

5: Send Notify Message : Non-Queued message

**What is window? How many components it's having write the name of components of window?**

**Answer:**      **Page no : 61**

Microsoft Windows consists of three important components. These are:

1. Kernel
2. GDI (Graphics Device Interface)
3. User

**How can we select the specific msg from queue**

**Answer:**      **ch#9 Page no : 88**

```
BOOL GetMessage()  
(  
LPMSG lpMsg,  
HWND hWnd,  
UINT wMsgFilterMin,  
UINT wMsgFilterMax  
)
```

**How can a combo box be added to the toolbar? Write appropriate function?**

**Answer:**

```
int CMainFrame::OnCreate(LPCREATESTRUCT lpCreateStruct)  
{  
    if (CFrameWnd::OnCreate(lpCreateStruct) == -1)  
        return -1;  
  
    if (!m_wndToolBar.CreateEx(this, TBSTYLE_FLAT, WS_CHILD |  
WS_VISIBLE | CBRSTOP  
        | CBRSGRIPPER | CBRSTOOLTIPS | CBRSTFLYBY |
```

```

CBRS_SIZE_DYNAMIC) ||
    !m_wndToolBar.LoadToolBar(IDR_MAINFRAME))
{
    TRACE0("Failed to create toolbar\n");
    return -1;    // fail to create
}

const int ID_COMBO_BOX = 2000;
if ( m_wndToolBar.CreateCombo( ID_COMBO_BOX_BTN,ID_COMBO_BOX ) )
{
    CComboBox& box = m_wndToolBar.GetCombo();
    box.AddString( "blue" );
    box.AddString( "green" );
    box.AddString( "yellow" );
    box.SetCurSel(1);
}

// addl init stuff follows...
}

```

**More details:** <http://www.drdoobs.com/windows/adding-a-combo-box-to-a-toolbar/184416845>

**What information about the resources is placed in .res?**

**Answer:** Page no 214

The resource-definition statements define the resources that the resource compiler puts in the resource (.Res) file. After the .Res file is linked to the executable file, the application can load its resources at run time as needed.

**How can a dialog be centered? Explain it with example and appropriate function call**

**Answer:**

An application can retrieve the dimensions of the owner window, calculate a new position that centers the dialog box in the owner window, and then set the position by using the [SetWindowPos](#) function.

**Explain the basic difference(s) between a message box and a dialog box.**

**Answer:**                      **Page no :248**

A dialog box is a temporary window an application creates to retrieve user input. An application typically uses dialog boxes to prompt the user for additional information for menu items.

A message box is a special dialog box that an application can use to display messages and prompt for simple input.

**Explain why all the application related messages come directly to the modal dialog box if it is active and not through its owner window's window procedure?**

**Answer:**                      **Page no: 249**

When creating the modal dialog box, the system makes it the active window. System disables the owner window, all mouse input is lost .To process messages for the modal dialog box, the system starts its own message loop, taking temporary control of the message queue for the entire application.

**What are the benefit(s) of declaring a resource as DISCARDABLE? 3 marks**

**Answer:**

To specify that it is okay to unload an unused resource from memory, you may list the **DISCARDABLE** keyword with the resource. DISCARDABLE resources allow more efficient memory usage

**We have an application in which there is a function named “TestingFunction”. There are 2 local integer variables declared in it. One is named “simpleVariable” and other is named “staticVariable”. The storage class of “staticVariable” is static. Suppose this function is called and when it returns the value of “simpleVariable” is lost but that of “staticVariable” is preserved. Why**

**is it so?**

**Answer:**

Static variables can be 'seen' within all functions in the source file. The static storage class specifies keeps the variable from being reinitialized each time the block, where the variable is defined, runs. The static variable is not destroyed on exit from the function; instead its value is preserved, and becomes available again when the function is next called.

**We have 128MB RAM in our system. Suppose we run a Win32 application that declares an array of 800MB whose each element is of 512KB. How will operating system handle the situation when the application requests 10 elements of that array?**

**Answer: #not sure**

If we want to access the first element of the second row then we have to jump as many numbers as the number of columns in the first row. This fact becomes important when we are passing arrays to functions. In the receiving function parameters, we have to write all the dimensions of the array except the extreme-left one. When passing arrays to functions, it is always call by reference by default; it is not call by value as in the default behavior of ordinary variable

**Suppose we have 2 menu items named "Show" and "Clear". Their identifiers are ID\_FILE\_SHOW and ID\_FILE\_CLEAR respectively. Write down the resource statements to add accelerators for these menu items5 marks**

**Answer: not sure**

```
IDR_ACCELERATOR ACCELERATORS DISCARDABLE

BEGIN "P", ID_TIMER_STOP, VIRTKEY, CONTROL, NOINVERT

"S", ID_TIMER_START, VIRTKEY, CONTROL, NOINVERT

"X", ID_FILE_EXIT, VIRTKEY, ALT, NOINVERT

END

ID_FILE_SHOW MENU DISCARDABLE

BEGIN

POPUP "&File"

BEGIN

MENUITEM "E&xit\tAlt+X", ID_FILE_EXIT END

POPUP "&Timer"
```

```

BEGIN

MENUITEM "&Start\tCtrl+S", ID_TIMER_START

MENUITEM "Sto&p\tCtrl+P", ID_TIMER_STOP, GRAYED

END

ID_FILE_CLEAR MENU DISCARDABLE

BEGIN

POPUP "&File"

BEGIN

MENUITEM "E&xit\tAlt+X", ID_FILE_EXIT END

POPUP "&Timer"

BEGIN

MENUITEM "&Start\tCtrl+S", ID_TIMER_START

MENUITEM "Sto&p\tCtrl+P", ID_TIMER_STOP, GRAYED

END

END

```

**An application whose primary thread's entry point is WinMain creates another thread using CreateThread function. Security Attribute parameter is set to NULL, stack size is 512KB and thread function is MyFunction. What would be the entry point and exit point of this secondary thread**

**Answer:**            **Page no : 345**

For creating thread we can use following functions:

`_beginthread()` and `_endthread()`

### Year 2012

#### **Owner Drawn Menu Items?**

**Answer:**    [Page no 241](#)

An application can completely control the appearance of a menu item by using an owner drawn item. Owner-drawn items require an application to take total responsibility for drawing selected (highlighted), selected, and cleared states.

**I have a server and a client program on the same machine. The server is a part of application it can start and stop arbitrarily. When the server is up. I want the client to connect to the server listening socket. There are win32 function to wait on file system changes(Read Directory Changes W) and registry changes (RegNotifyChangeKeyValue)- Write the names of related API's that we need to solve problem?**

**Answer:**

There is no such Win32 API, however this can be easily accomplished by using an event. The client would wait on that event to be signaled. The server would signal the event when it starts up.

The related API that you will need to use is CreateEvent, OpenEvent, SetEvent, ResetEvent and WaitForSingleObject.

**Is there a function that will freeze window for repainting for some time, while I do changes to the layout of the dialog? Write sample code.**

**Answer:**

If you find that you actually need to do this, you should send the window a [WM\\_SETREDRAW](#) message with the wParam set to FALSE. This indicates that the window should not be redrawn after its contents are changed.

When you want to re-enable drawing, send another WM\_SETREDRAW message, this time with the wParam set to TRUE.

Sample code:

```
// Disable window updates
```

```
SendMessage(hWnd, WM_SETREDRAW, FALSE, 0);

// Perform your layout here
// ...

// Re-enable window updates
SendMessage(hWnd, WM_SETREDRAW, TRUE, 0);
```

**What is meant by “Static Content” ?**

**Answer:**    **Page no 411**

Server blindly dumps HTML files to the clients. This is ‘static content’. Server reads file and modifies its output e.g. %%time%% replaced with current system time

**Write any six types of windows resources?**

**Answer:**    **Page no 214**

- Accelerator
- String Table
- Icon
- Bitmap
- Dialog
- Menu
- Cursor
- Version

**A university has 3 hostel building. Every Hostel have 50 rooms. In each room 3 Students at Max must be accommodated. Declare an array of integer type having name “ABC” to store the information above.**

**Answer:**    **Page no 15**

```
int ABC[3][50][3];
```



**We have a DLL that has been using by many applications We want to share a variable declared in that DLL among all these applications. How to do this?**

**Answer:**    [Page no 340](#)

Variables can be shared across multiple processes by making the separate data section as following.

```
#pragma data_seg([ [ { push| pop },] [ identifier,] ] [ "segmentname" [, "segment-class" ] ] )
```

**What is the purpose of Pager control?**

**Answer:**    [Page no 314](#)

A pager control is a window container that is used with a window that does not have enough display area to show all of its content.

**Q. What is the Mean by Resource definition Statement?**

**Answer:**    [Page no 214](#)

The resource-definition statements define the resources that the resource compiler puts in the resource (.Res) file. After the .Res file is linked to the executable file, the application can load its resources at run time as needed. All resource statements associate and identifying name or number with a given resource.

**Q. Why static variables are not destroyed when function returns?**

**Answer:**

A static can only be accessed from the function in which it was declared, like a local variable. The static variable is not destroyed on exit from the function, instead its value is preserved, and becomes available again when the function is next called. Static variables are declared as local variables, but the declaration is preceeded by the word static.

```
static int counter;
```

Static variables can be initialized as normal, the initialization is performed once only, when the program starts up.

**Q. How Transfer-Encoding is implemented in Exchange-2000.**

**Answer:**

Routing group boundaries and SMTP target destinations determine how Exchange 2000 Server encodes mail. Exchange 2000 will encode as quoted-printable or 7-bit or Transport-Neutral Encapsulation Format (TNEF) when sending between two servers/recipients in different routing groups, and to the Internet.

Exchange 2000 Server will encode in Binary or Summary TNEF when sending to a recipient/server in the same routing group.

**Q. How do I restrict my windows so it can't be resized longer or smaller then a certain size?**

**Answer:**

You restrict your window's size by handling the WM\_GETMINMAXINFO message. Your handler receives a pointer to a MINMAXINFO struct, which you fill in with the minimum and/or maximum size for your window.

**Q. If you include CS\_DBLCLKS in your window call style, the windows procedure receives which message for double click. ( Specify exact sequence).**

**Answer:** **Page no:209**

A double-click message is always the third message in a four-message series. The first two messages are the button-down and button-up messages generated by the first click. The second click generates the double-click message followed by another button-up message. For example, double-clicking the left mouse button generates the following message sequence:

- WM\_LBUTTONDOWN
- WM\_LBUTTONUP
- WM\_LBUTTONDOWNBLCLK
- WM\_LBUTTONUP

**Q. Write a recursive function definition for the following function.**

```
Int square (int n);
```

```
// Precondition : n>1
```

```
// Return the sum of square of number 1 through n
```

For example square of 3 returns 14 because  $1^2 + 2^2 + 3^2$  is 14

**Answer:**

```
public static int squares(int n)
```

**What is Synchronization Object? (2 marks)**

**Answer:**

**Page no :351**

Synchronization objects prevent other threads to use the shared data at the same. To synchronize access to a resource, use one of the synchronization objects in one of the wait functions. The state of a synchronization object is either signaled or non-signaled. The wait functions allow a thread to block its own execution until a specified non-signaled object is set to the signaled state.

**Briefly define Repeat Count? (2 marks)**

**Answer:**

**Page no:195**

You can check the repeat count to determine whether a keystroke message represents more than one keystroke. The system increments the count when the keyboard generates WM\_KEYDOWN or WM\_SYSKEYDOWN messages faster than an application can process them.

**Define property sheet? (2 marks)**

**Answer:**

**Page no: 311**

Another important dialogs are called property sheets. This property sheet enables you to select any category from the tabs.

**Briefly explain "Stock Objects". How an application gets handle to Stock Object? (3marks)**

**Answer:**

**Page no: 142**

During initialization, GDI creates a number of predefined objects that any application can use. These objects are called stock objects. An application calls the GetStockObject function to get a handle to a stock object, and the returned handle is then used as a standard object handle.

**Differentiate GetAsyncKeyState() and GetKeyState() functions? (3 marks)**

**Answer:**

**Page no: 198**

The application can use the GetKeyState function to determine the status of a virtual key at the time the current message was generated; it can use the GetAsyncKeyState function to retrieve the current status of a virtual key.

**Explain "Resource Only DLLs". (3 marks)**

**Answer:** Page no: 343

Resource Only DLL contains only resource of different language and local types. Resource only DLLs do not contain Entry Point or any DllMain Function. Use of resource-only DLL is for internationalization.

**Define Mutex Object and why it is used? (3 marks)**

**Answer:** Page no: 343

Two or more processes can call CreateMutex to create the same named mutex. The first process actually creates the mutex, and subsequent processes open a handle to the existing mutex.

Using Named Mutex object you can check the application instance whether it is already running or not. Recreating the named mutex open the previous mutex object but set last error to ERROR\_ALREADY\_EXISTS.

**Suppose an application has a window and 2 menu resources, one is IDM\_MENU1 and second is IDM\_MENU2. While registering the window class in this application, we make IDM\_MENU1 the class menu. Now, we have 2 scenarios:**

**We give the menu handle of IDM\_MENU2 during creating the window.**

**We give NULL as the menu during creating the window.**

**You have to tell which menu will be associated to the window in both cases. (5 marks)**

**Answer:** Page no 231

IDM\_MENU2 will be associated to the window in both cases because hMenu is the handle of the menu IDM\_MENU2 if the handle of the menu is specified then this will override class window menu that is IDM\_MENU1. Whereas the second reason concern i do not understand that whether sir is giving null value as return value or null value in lpMenuName parameter.

**How can I return multiple values from a function? Illustrate the concept using C++ code (5 marks)**

**Answer:**

```
#include<iostream.h>
#include<conio.h>
void calc(int x, int y, int *add, int *sub)
```

```

{
    *add = x+y;
    *sub = x-y;
}

void main()
{
    int a=20, b=11, p,q;
    clrscr();
    calc(a,b,&p,&q);
    cout<<"Sum = "<<p<<" Sub = "<<q;
    getch();
}

```

**I've allocated a console in my GUI program. When I try to close the console window, it hangs around for a while. Why? (5 marks)**

**Answer:**

While only one console can be associated with a process, it's possible for more than one process to share the same console. When you call `FreeConsole()`, Windows detaches your app from the console, and will close it if no other processes are using it. There is a difference between NT and 9x in when the close is performed. NT seems to check the status immediately and close the console. Windows 9x seems to take more of a "garbage collection" approach, preferring to wait until some action is performed on the console widow before checking if it's time to get rid of it.

**MIME stands for? (1 Mark)**

**Answer:**      **Page no : 395**

MIME stands for Multi-purpose Internet Mail Extensions.

**CGI stands for? (1 Mark)**

**Answer:**      **Page no : 412**

CGI is Common Gateway Interface. Win32 executable execute by the server.

**Write some operations of Sockets (2 Marks)**

**Answer:**      **Page no : 371**

The following are the basic operations performed by both server and client systems.

1. Create an unbound socket
2. Binding Server
3. Connecting Client
4. Listen
5. Accept
6. Send
7. Receive

**Copy on write protection (3 Marks)**

**Answer:**      **Page no: 300**

Copy-on-write protection is an optimization that allows multiple processes to map their virtual address spaces such that they share a physical page until one of the processes modifies the page. This is part of a technique called lazy evaluation, which allows the system to conserve physical memory and time by not performing an operation until absolutely necessary.

**Write any 5 features of DOS Programming (3 Marks)**

**Answer:**      **Page no : 2**

- It "owns" the system
- Provides direct device access
- Non-portability across machines
- Status polling
- No multitasking

**How windows socket function works? (3 Marks)**

**Answer:**      **Page no : 371**

Windows Socket defines a standard service provider interface (SPI) between the application programming interface (API), with its exported functions and the protocol stacks. The following are the basic operations performed by both server and client systems.

1. Create an unbound socket
2. Binding Server
3. Connecting Client
4. Listen
5. Accept
6. Send
7. Receive

**There is a text file that I want to embed as resource so I can use it when needs how I can do it? (3 Marks)**

**Answer:**

You embed your text file in a resource file like this:

nameID typeID filename

where nameID is some unique 16-bit unsigned integer that identifies the resource and typeID is some unique 16-bit unsigned integer greater than 255 that identifies the resource type (you may define those integers in the resource.h file). filename is the path to the file that you want to embed its binary contents into the compiled resource.

So you might have it like this:

In resource.h:

// Other defines...

```
#define TEXTFILE    256
#define IDR_MYTEXTFILE 101
```

**What is the purpose of Bind Function?**

**Answer:**      **Page no : 377**

Bind:

The bind function associates a local address with a socket.

```
int bind(SOCKET s,  
  
const struct sockaddr* name,  
  
int namelen);
```

**What is the purpose of List View Control?**

**Answer:**      **Page no : 313**

A list-view control is a window that displays a collection of items. The control provides several ways to arrange and display the items.

**What is a custom resource?**

**Answer:**      **Page no : 214**

An application defined resource, also called a custom resource, contains any data required by a specific application.

**Write following statement in words**

**Const char \*ptr=buff**

**Answer:**      **Page no : 56**

Here, ptr has been declared as “variable pointer to constant data”. In this case, the data to which the ptr is pointing to remains constant and cannot be modified after initialization.

The contents of ptr (address) are variable and we can change the contents of ptr. **Differentiate between Simple Windows Programs & Dynamic Link Library program?**



**Answer:**      **Page no :**

A dynamic-link library (DLL) is a module that contains functions and data that can be used by another module (application or DLL).

In 32-bit windows programming, we are freed from the curse of 64k segments, far and near pointers, 16-bit integers and general limitations.

**What are the three conditions of sending WM\_Paint Message?**

**Answer:**      **Page no : 167**

- Any hidden part of window becomes visible Window is resized (and CS\_VREDRAW, CS\_HREDRAW style bits were set while registering the window class).
- Program scrolls its window.
- InvalidateRector InvalidateRgnis called by the application.

**Write down any three parameter of send function?**

**Answer:**

HWND hWnd,  
UINT Msg,  
WPARAM wParam,  
LPARAM lParam

**If a network Engineer wants to send a packet to group of nodes then which IP address class will be used?**

**Answer:**      **Page no : 317**

ICC\_INET\_CLASSES:      Load IP address class.

**Write down message sequence of Shift +S to windows procedure?**

**Answer:**      **Page no : 199**

Whenever the user presses the hot key, the system sends a WM\_SYSCOMMAND message specifying SC\_HOTKEY.

**What is the equivalence in pointers and arrays in C?**

**Answer:**      **Page no : 9**

An array name is actually a pointer to the first element of the array. When you declare an array, the name is a pointer. You cannot alter the value of this pointer.

**Briefly explain Checkbox?**

**Answer:**      **Page no : 289**

A check box consists of a square box and application-defined text (label), an icon, or a bitmap, that indicates a choice the user can make by selecting the button. A check box can be one of four styles: standard, automatic, three-state, and automatic three-state.

**Use of Translate message and Dispatch message. (5 marks)**

**Answer:**      **Page no : 100**

**Translate message :** A thread's message loop must include TranslateMessage if the thread is to receive character input from the keyboard. To retrieve this value, the message loop must contain TranslateMessage, which translates the virtual-key message into a character message (WM\_CHAR) and places it back into the application message queue.

**Dispatch message:** The DispatchMessage function sends a message to the window procedure associated with the window handle specified in the MSG structure. If the window handle is HWND\_TOPMOST, DispatchMessage sends the message to the window procedures of all top-level windows in the system. If the window handle is NULL, dispatchMessage does nothing with the message.

**Responsibilities of Kernel. (5 marks)**

**Answer:**      **Page no : 61**

Kernel is a main module of the operating system. This provides system services for managing threads, memory, and resources.

Kernel has to perform very important responsibilities e.g.

1. Process Management
2. File Management
3. Memory Management (System and Virtual Memory)

**How to send message? example. (3 marks)**

**Answer:**      **Page no :**

The SendMessage function sends the message to the window procedure corresponding to the given window. The function waits until the window procedure completes processing and then returns the message result.

**Disadvantages of threads. (3 marks)**

**Answer:**      **Page no : 345**

Threads major disadvantage is that they make the system slow because thread uses the time sharing concept that is another name multitasking

**What is handle to menu? (2 marks)**

**Answer:**      **Page no : 230**

The system generates a unique handle for each menu. A menu handle is a value of the HMENU type. An application must specify a menu handle in many of the menu functions. We can retrieve handle of a menu using **getMenu**.

**Which function is used to create window class? (2 marks)**

**Answer:**      **Page no :**

CreateWindow or CreateWindowEx is used

**There was question regarding semaphore. (2 marks)**

**Answer:**      **Page no : 357**

Semaphore maintains a count between zero and some maximum value, limiting the number of threads that are simultaneously accessing a shared resource.

### **Year 2010**

**Briefly explain Message Box.**

**Answer:**      **Page no : 117**

A message box is a special dialog box that displays a note, caution, or warning to the user. For example, a message box can inform the user of a problem the application has encountered while performing a task. For more information, see Message Boxes.

**What are the uses of Dialog.**

**Answer:**      **Page no :**

An application uses a dialog box to prompt the user for input needed to complete a command.

**Thread Synchronization?**

**Answer:**      **Page no : 351**

Using threads we can use lot of shared variables. These shared variables maybe used by a single thread further more these variables may also be used and changed by several parallel threads. If there are several threads operating at the same time then a particular DC handle can be used in one of the threads only. If we want to use a single DC handle in more than one thread, we use synchronization objects. Synchronization objects prevent other threads to use the shared data at the same.

**Any two advantages of threads?**

**Answer:**      **Page no : 345**

1. Threads can be used to start another activity parallel. E.g. saving file on disk, automatically while you are typing.

2. Perform different calculations parallel.

1 question is about to write a program with DLL export function which prints a message "Hello, DLL exported...(5 marks)

Answer: not sure

```
library TestLibrary;
```

```
uses SysUtils, Classes, Dialogs;
```

```
procedure DllMessage; export;
```

```
begin
```

```
  ShowMessage('Hello world from a Delphi DLL') ;
```

```
end;
```

```
exports DllMessage;
```

```
begin
```

```
end.
```

1 is about a mouse center button, and how we check is there any mouse attached to system....(5 marks)

Answer: Page no : 206

An application can determine whether the system includes a mouse by passing the SM\_MOUSEPRESENT value to the GetSystemMetrics function. An application can determine the number of buttons on the mouse by passing the SM\_CMOUSEBUTTONS value to the GetSystemMetrics function.

What is Virtual Directory? and explain...(5 marks);

3 steps of resource file....(3 marks)

Answer: Page no : 217

1. .rc File (text file containing resource statements)
2. Compile to .res file (using resource compiler)
3. Link with other files to make final EXE (using linker) File in windows.

What is list box (2 marks)

Answer: Page no : 294

List box is a collection of different items. List box items can be represented by text strings, bitmaps, or both. If the list box is not large enough to display all the list box items at once, the list box provides a scroll bar.

**What is a window menu briefly explain?2marks**

**Answer:**      **Page no : 229**

The Window menu (also known as the System menu or Control menu) is a pop-up menu defined and managed almost exclusively by the operating system. The user can open the window menu by clicking the application icon on the title bar or by right clicking anywhere on the title bar.

**Difference between shortcut menu and window menu 5marks**

**Answer:**      **Page no : 229**

Shortcut menu	Window menu
A shortcut menu is not attached to the menu bar; it can appear anywhere on the screen.	A window menu can be opened by clicking the application icon on the title bar or by right clicking anywhere on the title bar.
It is defined and managed by application.	It is defined and managed almost exclusively by the operating system

**Briefly explain DNS?5marks**

**Answer:**      **Page no : 370**

Domain Name System (DNS), the locator service of choice in Microsoft® Windows®, is an industry-standard protocol that locates computers on an IP-based network. IP networks such as the Internet and Windows networks rely on number-based addresses to process information.

**What is dialog box menu 3marks**

**Answer:**      **Page no : 248**

It is a window menu. WM\_SYSMENU is used in dialog box template to specify this menu.

**YEAR 2008**

**What is a process?**

**Answer:**      **Page no : 327**

A running application that consists of a private virtual address space, code, data, and other operating-system resources, such as files, pipes, and synchronization objects that are visible to the process. A process also contains one or more threads that run in the context of the process. A process, in the simplest terms, is an executing program

**What is HTTP?**

**Answer:**      **Page no : 395**

HTTP is a Stateless protocol.

- No information or “state” is maintained about previous HTTP requests
- Easier to implement than state-aware protocols

**Briefly define Modal Loop?**

**Answer:**      **Page no : 251**

Modal loop is run by Modal dialogs and process message as does application message loop.

**Define briefly Group Box?**

**Answer:**      **Page no :**

**Which are the three parts of name when using internet address family?**

**Answer:**      **Page no :**

ICC\_INET\_ADDRSTRUC

**Define the names of resource-definition statements' categories?**

**Answer:**      **Page no : 214**

1. Controls
2. Statements

### 3. Resources

**State three differences between a Window Procedure and a Dialog Procedure?**

**Answer:**      **Page no :**

Dialog Procedure	Window Procedure
The DialogProc processes messages sent to a modal or modeless dialog box.	The window procedure processes messages for all windows of the class.
Dialog Procedure did not call the DefWindowProc function to process unwanted messages.	Unwanted messages are processed internally by the dialog box window procedure.
Window Procedure is associated with a window class.	Dialog procedure is used with dialog box class.

**Define the purpose of connect function with syntax. What does it return?**

**Answer:**      **Page no : 381**

The connect function establishes a connection to a specified socket.

```
int connect(SOCKET s, const struct sockaddr* name, int namelen);
```

**Define Dynamic Link Libraries in detail? Also explain its relation with memory management?**

**Answer:**      **Page no : 332**

A dynamic-link library(DLL) is a module that contains functions and data that can be used by another module (application or DLL).

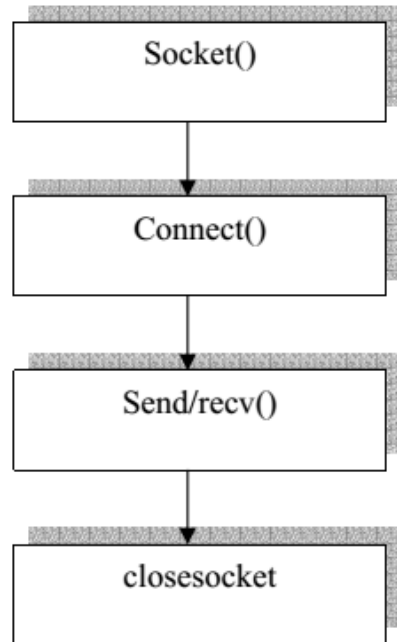
Every process that loads the DLL maps it into its virtual address space. After the process loads the DLL into its virtual address, it can call the exported DLL functions. The system maintains a per-thread reference count for each DLL. When a thread loads the DLL, the reference count is incremented by one. When the process terminates, or when the reference count becomes zero



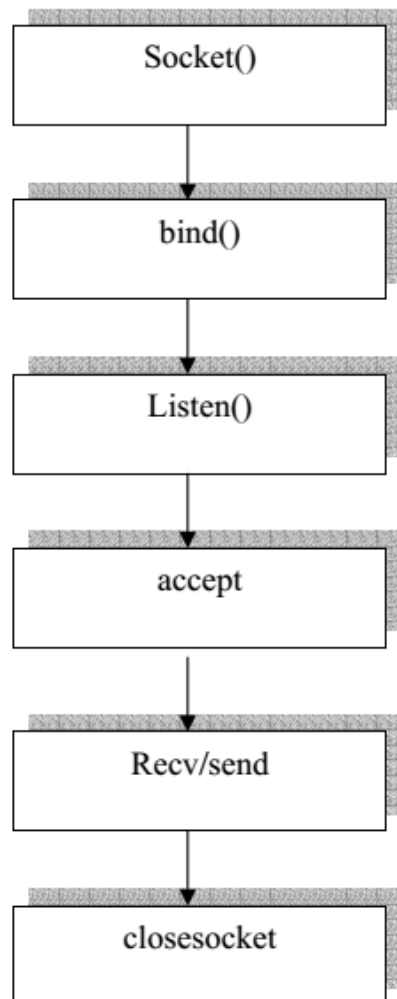
(run-time dynamic linking only), the DLL is unloaded from the virtual address space of the process.

**Show the flow of server and client socket calls using block diagram?**

**Answer:**      **Page no : 386**



**Figure 1 Client Connection**



## REVIEW FROM PREVIOUS MIDTERM PAPERS

1.-----is used to check the predefined identifiers.

1. #include

2. #ifdef **ANS**

3. #def

4. #elif

Microsoft Windows consists of three important components. These are:

1. Kernel

2. Pointer

3. User

Option

a. I

b. li

c. I and iii **Ans**

d. I AND ii

-----function to convert the message into character messages.

**Ans: TranslateMessage**

If you pass WS\_CAPTION style to create window(),-----style will be included automatically in it.

**Ans WS\_BORDER style**

WM\_PAINT tells the window procedure that the window's client area has changed and must be repainted

Q:1 Define the structure of PAINTSTRUCT

**Answer:** **chapter no : 14 , Page # 168**

The PAINTSTRUCT structure contains information for an application. This information can be used to paint the client area of a window owned by that application.

```
typedef struct tagPAINTSTRUCT {
```

```
HDC hdc; //Handle to the Device context
```

```
BOOL fErase; /*erase back ground of this parameter is true*/
```

```
RECT rcPaint; /*rectangle to the invalidate region*/
```

```

BOOL fRestore;

BOOL fIncUpdate; //updatation true/false

BYTE rgbReserved[32]; //rgb values

} PAINTSTRUCT, *PPAINTSTRUCT;
Q2: Define window according MICROSOFT.

```

**Answer:**                    **chapter no : 1 , Page # 3**

On November 10, 1983, Microsoft announced Microsoft Windows, an extension of the MS-DOS® operating system that would provide a graphical operating environment for PC users. Microsoft called Windows 1.0 a new software environment for developing and running applications that use bitmap displays and mouse pointing devices. With Windows, the graphical user interface (GUI) era at Microsoft had begun.

**Q3: Define Macro types with example**

**Answer:**                    **chapter no : 5 , Page # 30**

A macro is a fragment of code which has been given a name. Whenever the name is used, it is replaced by the contents of the macro. There are two kinds of macros. They differ mostly in what they look like when they are used. **Object-like macros** resemble data objects when used, **function-like macros** resemble function calls. For example, here's a macro that computes the maximum of two numeric values:

```
#define min(X, Y) ((X)>(Y) -> (X):(Y))
```

**Q4: write the structure of CREATWINDOW and tell about parameters which are used in ?**

**Answer:**                    **chapter no : 9 , Page # 80**

```

HWND CreateWindow(

LPCTSTR lpClassName; //class name (identification)

LPCTSTR lpWindowName; //Window caption bar Name

DWORD dwStyle; // style of the windows

Int x; //starting X point of window on screen

```

```

Int y;    //starting Y point of window on screen

Int width; //Width of the window from starting point

Int height; //height of the window from starting Y point

HWND hWndParent; //handle the parent window if any

HMENU hMenu; // handle the Menu if any

HINSTANCE hInstance; //handle of the instance LPVOID lpParam; //void parameter

);

```

**Q5: define virtual key function msg?**

**Answer:**                **chapter no : 10 , Page # 100**

A virtual-key message contains a virtual-key code that identifies which key was pressed, but not its character value. To retrieve this value, the message loop must contain TranslateMessage, which translates the virtual-key message into a character message (WM\_CHAR) and places it back into the application message queue

**Q6 : Define typedef**

**Answer:**                **chapter no : 6 , Page # 38**

A typedef declaration lets you define your own identifiers that can be used in place of type specifiers such as int, float, and double. The names you define using typedef are NOT new data types. They are synonyms for the data types or combinations of data types

they represent. The following statements declare LENGTH as a synonym for int, then use this typedef to declare length, width, and height as integral variables.

```
typedef int LENGTH;
```

```
LENGTH length, width, height;
```

**q3:steps to handle message generate against any action**

**Q7: Define clipboard**

**Answer:**                **chapter no : 8 , Page # 65**

- User32.dll manages clipboard.
- Clipboard is used to cut copy and paste operations.

- Clipboard is temporary storage area. When you shut down windows, data saved in clipboard will be lost.

**Q8: Write at least two responsibility of GDI?**

**Answer:** chapter no : 8 , Page # 61

GDI is a subsystem responsible for displaying text and images on display devices and printers.

**Q9: In 32 bit version of the Microsoft wind, when a window is created, which structure is used by operating system?**

**Answer:** chapter no : 9 , Page # 80

```
HWND CreateWindow(  
    LPCTSTR lpClassName; //class name (identification)  
    LPCTSTR lpWindowName; //Window caption bar Name  
    DWORD dwStyle; // style of the windows  
    Int x; //starting X point of window on screen  
    Int y; //starting Y point of window on screen  
    Int width; //Width of the window from starting point  
    Int height; //height of the window from starting Y point  
    HWND hWndParent; //handle the parent window if any  
    HMENU hMenu; // handle the Menu if any  
    HINSTANCE hInstance; //handle of the instance LPVOID lpParam; //void parameter  
);
```

**Q10 : Program executed or not .Give reason .code given below**

```

#include<iostream.h>
#include<conio.h>
Const int z=6
Void sum(int a,int b)
{
Z=a;
Coutz+b; }
Int main()
{
Sum(5,9);
Getch();
}

```

**Answer:** code isn't complete

**Q11 .Can we pass argument to Macro? Explain with example? 5 marks**

**Answer:** chapter no : 5 , Page # 30

To define a macro that takes arguments, you use the #define command with a list of parameters in parentheses after the name of the macro. The parameters may be any valid C identifiers separated by commas at the top level (that is, commas that aren't within parentheses) and, optionally, by white-space characters. The left parenthesis must follow the macro name immediately, with no space in between. For example, here's a macro that computes the maximum of two numeric values:

```
#define min(X, Y) ((X)>(Y) ? (X):(Y))
```

**Q 12:**

**Separate the system window class that can be used by the user process from system window classes that only use by the system**

- |               |                |
|---------------|----------------|
| i.Button      | user processes |
| ii.DDEMLEvent | system         |
| iii.Mesage    | system         |
| iv. ScrolBar  | user processes |
| v. ComboLBOx  | system         |

**Q 13: What is window ? how many components its have write the name of components of window**

**Answer:** chapter no : 1 , Page # 3

On November 10, 1983, Microsoft announced Microsoft Windows, an extension of the MS-DOS® operating system that would provide a graphical operating environment for PC users. Microsoft called Windows 1.0 a new software environment for developing and running applications that use bitmap displays and mouse pointing devices. With Windows, the graphical user interface (GUI) era at Microsoft had begun.

Microsoft Windows consists of three important components. These are:

1. Kernel
2. GDI (Graphics Device Interface)
3. User

**Q.14: How can we select the specific msg from queue**

**Answer:** [ch#9 Page no : 88](#)

```
BOOL GetMessage()  
(  
LPMSG lpMsg,  
HWND hWnd,  
UINT wMsgFilterMin,  
UINT wMsgFilterMax  
)
```

**Q.15: Write three conditions in which paint msg sent?**

**Answer:**

- Any hidden part of window becomes visible Window is resized (and CS\_VREDRAW, CS\_HREDRAW style bits were set while registering the window class).
- Program scrolls its window.
- Invalidate Rector InvalidateRgnis called by the application.



**Q.16: Write the syntax of send message function and also write what will happen when send message function is used?**

**Answer:** Chapter:10 Page no : 103

An application typically sends a message to notify a window procedure to perform a task immediately. The **SendMessage** function sends the message to the window procedure corresponding to the given window. The function waits until the window procedure completes processing and then returns the message result. Parent and child windows often communicate by sending messages to each other.

**Q.17: Write the name of the two type of implicit? 2 marks**

**Answer:** Chapter:6 Page no : 39

Implicit type casting (coercion) is further divided in two types

- Promotion
- Demotion

**Q.18: Super classing se related ek question jo scenario based tha**

**Answer:** #not sure as q is not so clear

After implementation of Sub-classing or Super Classing don't forget to call window procedure function.

**Q.19: Write a 5 line code for handle when the message is.....bhool gaya :P**

**Answer b bhol gya :D**

**Q.20: Code for window procedure while creating window**

**Answer:**

Repeated

**Q.21: Type of brushes and briefly explain?**

**Answer:** Chapter#8, Page no : 63

There are two types of brushes: logical and physical. A logical brush is one that you define in code as the ideal combination of colors and/or pattern that an application should use to paint

shapes. A physical brush is one that a device driver creates, which is based on your logical-brush definition.

**Q.22: program to write in c/c++.of multiplication 4 integer taken as parameter and taken as argument . Also make two macro of Multiplication 4 integer (5 Marks )**

**Answer: #not sure about macro stuff**

```
#include<iostream.h>

#include<conio.h>

#define mul(a, b)(c=a*b)

#define mul1(x, y)(d=x*y)

void main()

{

int a,b,mul1,x,y,mul;

cout<<"Enter the first no."<<endl;

cin>>a;

cout<<"Enter the second no."<<endl;

cin>>b;

cout<<"Enter the 3rd no."<<endl;

cin>>x;

cout<<"Enter the 4th no."<<endl;

cin>>y;

mul=a*b*x*y;

cout<<"multiplication of four numbers are "<<mul;

getch();

}
```

**Q.23: SubClassing respond at which three manners when got message . (3 marks)**

**Answer:** Chapter#12 Page no : 129

**When application subclasses a window, it can take three actions with the message:**

- (1) Pass the message to the original window procedure;
- (2) Modify the message and pass it to the original window procedure;
- (3) Not pass the message.

**Q.24: Make a statement of "Variable pointer to const data "(2 Marks)**

**Answer:** Chapter :7 Page no : 56

- Variable pointer to Constant data:

```
const char * ptr = buff. //variable pointer to constant data
```

```
*ptr = 'a'; // it will be an error ptr = buf2;
```

Here, ptr has been declared as "variable pointer to constant data". In this case, the data to which the ptr is pointing to remains constant and cannot be modified after initialization.

The contents of ptr (address) are variable and we can change the contents of ptr

**Q.25: Scenario was given of three Multi Dimension array . need to declare that 3 kind of data in Array (Marks 3)**

**Answer:** Chapter#3 Page : 15

```
int provinces[50][500][1000];
```

```
// This will declare a three dimensional array.
```

**Q.26: How do i translate GetLastError()into string ? (5 marks )**

**Answer:** Chapter# 9 Pageno: 89

To get extended error information, use GetLastError function.

**Q.27:** .I have a tab-based application for windows. I would like to add a subtle gradient to the back ground of my tab control. How would I go around doing this? What is the best method for me to Use? 5 marks

**Answer:** #internet

To use GDI you'll need the GradientFill function. You can also use GDI+ to get gradients. Here's a plain GDI example:

```
TRIVERTEX    vert[2];
GRADIENT_RECT gRect;
vert[0].x     = 0;
vert[0].y     = 0;
vert[0].Red   = 0x0000;
vert[0].Green = 0x0000;
vert[0].Blue  = 0x0000;
vert[0].Alpha = 0x0000;

vert[1].x     = 100;
vert[1].y     = 32;
vert[1].Red   = 0x0000;
vert[1].Green = 0x0000;
vert[1].Blue  = 0xff00;
vert[1].Alpha = 0x0000;

gRect.UpperLeft = 0;
gRect.LowerRight = 1;
GradientFill(hdc,vert,2,&gRect,1,GRADIENT_FILL_RECT_H);
```

As for the tab control, you could sub-class the control and override its non-client and client drawing handlers to render the gradient.

To sub class a control, first create the control and then replace its WNDPROC function:

```
OldWndProc = (WNDPROC)SetWindowLongPtr (hControl, GWLP_WNDPROC,
(LONG_PTR)NewWndProc);
```

then, in your new WNDPROC:

```
NewWndProc (usual args)
{
    switch message
    {
        case paint:
            draw gradient
            return result

        default:
            return CallWindowProc (OldWndProc, ..args..); <- important!
    }
}
```

}

**Q.28 :** Write a program using WinMain function in which you will display a message box.  
Message Box display “**just Hello world program !**” in this title bar (title bar of the message box in given) 5.marks



**Answer:**

```
#define WIN32_LEAN_AND_MEAN
```

```
#include <windows.h>    // the main windows headers
```

```
#include <windowsx.h>    // a lot of cool macros
```

```
// main entry point for all windows programs
```

```
int WINAPI WinMain(HINSTANCE hinstance,
```

```
HINSTANCE hprevinstance,
```

```
LPSTR lpcmdline,
```

```
int ncmdshow)
```

```
{
```

```
// call message box api with NULL for parent window handle
```

```
MessageBox(NULL, "Hello World!",
```

```
"JUST ANOTHER HELLO WORLD PROGRAM",
```

```
MB_OK | MB_ICONEXCLAMATION);
```

```
// exit program  
return(0);  
  
} // end WinMain
```

**Q.29:** How an application get handle to Stock Object? 3 marks

**Answer:** Chapter#13 Pageno: 142

An application calls the GetStockObject function to get a handle to a stock object, and the returned handle is then used as a standard object handle.

**Q.30:** What is the use of “SendMessage()”and “Post Message()”functions? Also give the difference between them 3 marks

**Answer:** Chapter# 10 Pageno:103

**SendMessage():**

The SendMessage function sends the message to the window procedure corresponding to the given window. The function waits until the window procedure completes processing and then returns the message result.

**Post Message():**

An application typically posts a message to notify a specific window to perform a task. PostMessage creates an MSG structure for the message and copies the message to the message queue. The application's message loop eventually retrieves the message and dispatches it to the appropriate window procedure.

**Q.31:** Enlist two responsibilities of kernel? 2 marks

**Answer:** Chapter# 8 Page no: 61

Kernel is a main module of the operating system. This provides system services for managing threads, memory, and resources.

Kernel has to perform very important responsibilities e.g.

1. Process Management
2. File Management
3. Memory Management (System and Virtual Memory)

**Q.32: Some Times before selecting a new GDI object in any Device Context we keep previously selected Object stored ,why ? 2 marks**

**Answer:** Chapter# 13 Page no: 144

Common practice is to track the original object that was selected into the DC and **select it back** when all work is accomplished with the new object.

**2-Marks**

**Q.33: Define realization process?**

**Answer:** Chapter# 13 Page no: 144

Objects are converted from logical objects to physical objects using the realization process. Selecting a logical object into a DC involves converting the logical object into a physical object that the device driver uses for output. This process is called realization.

**Q.34: Create a Child window of type Button, with**

- Title "Select"
- Horizontal position: 70, vertical position: 150
- Width:150 height 200

**NOTE:** CreateWindow() function sy ik button type ki child window ka code likhna tha ye

**Answer: #notsure(really)**

HWND CreateWindow(

LPCTSTR lpClassName; //class name (identification)

LPCTSTR lpWindowName [in, title]//Window caption bar Name

DWORD dwStyle[WS\_CHILD]; // style of the windows

Int x[70]; //starting X point of window on screen

Int y[150]; //starting Y point of window on screen

Int nwidth[150]; //Width of the window from starting point

```
Int nheight[200]; //height of the window from starting Y point  
  
HMENU hMenu; // handle the Menu if any  
  
HINSTANCE hInstance; //handle of the instance  
  
LPVOID lpParam; //void parameter  
  
);
```

**Q.35:** according to \_\_cdecl calling convention

1. Argument Order : **Right to left**
2. Stack maintains responsibility : **Calling function pops up argument from the stack**
3. Name-decoration convention : **\_ underscore is used as prefix**

**Q.36:** Name any five elements include in application window?

**Answer:**                      **Page no: 79**

An application window includes elements such as a title bar, a menu bar, the window menu (formerly known as the system menu), the minimize button, the maximize button, the restore button, the close button, a sizing border, a client area, a horizontal scroll bar, and a vertical scroll bar.